

App. No. 09/820,501

**EXHIBIT A**

**BEST AVAILABLE COPY**

**Disclosure**

Prepared for and/or by an IBM Attorney - IBM Confidential

Created By: Nizam Ishmael Jr Created On:

Last Modified By: Nancy Werchan Last Modified On:

Required fields are marked with the asterisk (\*) and must be filled in to complete the form.

**\*Title of disclosure (In English)**

Workload Balancing GUI update algorithm

**Summary**

Status	Awaiting Search
Processing Location	AUS
Functional Area	52 - SDM - SOLUTION PROVIDERS MARKETING (R. C. Timpson)
Attorney/Patent Professional	Cynthia S Byrd/Austin/IBM
IDT Team	Clifford Spinae/Austin/IBM
Submitted Date	
Owning Division	SDM
Incentive Program	
Lab	
Technology Code	
PVT Score	No PVT score has been calculated. To calculate a PVT score, press the 'Calculate' button.

**Inventors with Lotus Notes IDs**

Inventors: Nizam Ishmael Jr/Austin/IBM, Mandeep S Sidhu/Austin/IBM @ IBMUS, Kulvir S Bhogal/Austin/IBM @ IBMUS, Bal Baweja/Austin/IBM @ IBMUS

Inventor Name	Inventor Serial	Div/Dept	Inventor Phone	Manager Name
Ishmael Jr, Nizam	919570	7N/XDIA	N/A	Gutierrez, Alfredo
Sidhu, Mandeep S.	806578	7N/Q5FA	N/A	Juliano, S.C. (Susan)
Bhogal, Kulvir Singh (Bicky)	0A5420	7J/B59A	N/A	Hicks, R.J. (Richard)
Baweja, Bal	234133	7N/XDIA	N/A	Gutierrez, Alfredo

&gt; denotes primary contact

**Inventors without Lotus Notes IDs****IDT Selection****\*Main Idea**

To view the main idea for this disclosure, click on this doctlink --&gt;

**\*Critical Questions (Questions 1-9 must be answered)**

Patent Value Tool (Optional - this may be used by the inventor and attorney to assist with the evaluation)

[REDACTED] Workload Balancing . . . update algorithm - continued

**Evaluation**

**Search Information**

**Search Office Information**

**Post Disclosure Text & Drawings**

[REDACTED]

[REDACTED]

[REDACTED] - continued

**Main Idea for Disclosure** [REDACTED]

Prepared for and/or by an IBM Attorney - IBM Confidential

Archived On: [REDACTED]

**Title of disclosure (In English)**

Workload Balancing GUI update algorithm

**Idea of disclosure**

1. Describe your invention, stating the problem solved (if appropriate), and indicating the advantages of using the invention.

Our invention is an algorithm to update a WorkLoad Balancing GUI that can be used for messaging applications such as MQSeries.

2. How does the invention solve the problem or achieve an advantage, (a description of "the invention", including figures inline as appropriate)?

The workload balancing GUI will help an administrator track messages that are sent to a messaging server. Our invention is the algorithm that will update this GUI.

An example GUI would be similar to manager.jpg (see attachment).

The algorithm assumes that an administrator will be logged into a messaging server and will be tracking the path/progress of received messages. The algorithm allows the messaging server to only track messages that it receives. This algorithm also assumes that the messaging server can resend the message to another messaging server (yet all the message status information will be relayed back to the originating messaging server.) The algorithm assumes that there will be two types of messages implemented: 1) tracking messages, which are used solely to for the GUI, 2) data messages that are the messages that are currently being used by messaging servers.

The flow chart for the algorithm is attached as algorithm.jpg (see attachment).

The following is a short description of the algorithm.

110. The Receiving server receives the message.

If the message received was a Data message then go to 113

For a Tracking message go to 112.

113. This step checks if the data message was sent from another messaging server, as opposed to being initiated by an application.

If yes the message is from another server then go to step 115.

If no the message is NOT from another messaging server go to step

114.

115. A tracking message is sent back to the messaging server that sent the message so that its GUI can be updated. Now go to step 114 so the Data message can be sent to its receiving Queue manager.

114. This step takes the Data message and sends it to a Queue Manager. Go to step 116.

116. The Queue Manager receives the message during this step. Now we go to step 124 and step 120.

124. A tracking message is sent back to the server i.e., the same server that the message is at to update

Main idea for disclosure [REDACTED] - continued

its GUI. Go to step 110 (server receives message)

120. This checks if the data message is at its final destination.

If yes the message is at its final destination and we go to step 121.

If no the message is NOT at its final destination and we go to step 122 and step 123.

121. A tracking message is sent back to the messaging server to update the GUI, informing the GUI that the message is at its final destination. We next go to step 110 with the tracking message.

122. The Data message is not at its final destination, so a tracking is sent message back to the server to inform it that the message is being sent to another messaging server. We next go to step 110 with the tracking message.

123. The Data message is NOT at its final destination, so it needs to be forwarded to the next Queue Manager. This is assuming that the server contains the destination queue. If it does not then the data message is sent to another messaging server.

112. Step 110 received a tracking message and then goes to step 112. In this step the GUI is updated. And we proceed to step 117.

117. This step checks if the tracking message needs to go to another messaging server.

If yes the tracking message needs to go to the next server in the chain to update its GUI. We then go to step 118.

If no the tracking message does not need to update other messaging server GUIs. Do nothing.

118. The tracking message needs to go to the next server in the chain to update its GUI. Accordingly, the message is sent to the next messaging server. This information about the next messaging server will be contained in the GUI.

3. If the same advantage or problem has been identified by others (inside/outside IBM), how have those others solved it and does your solution differ and why is it better?  
no

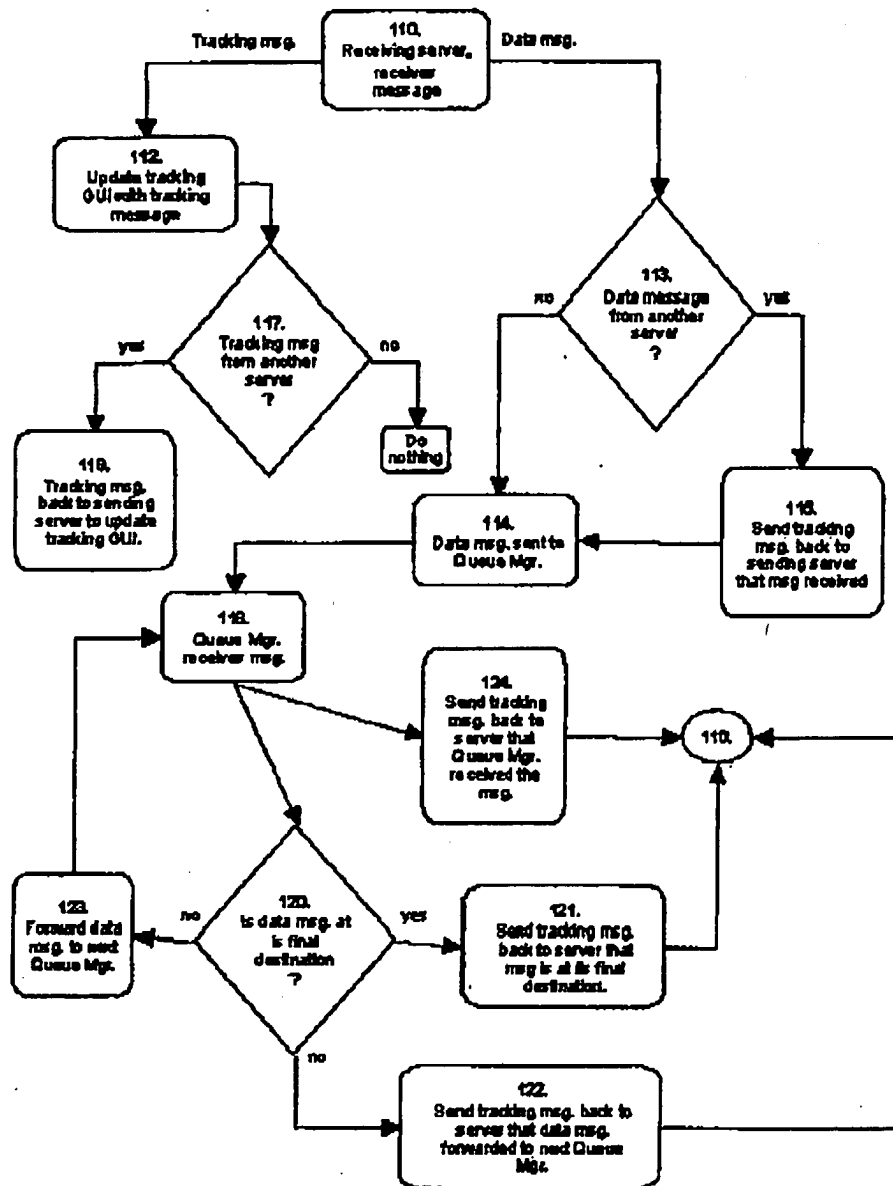
4. If the invention is implemented in a product or prototype, include technical details, purpose, disclosure details to others and the date of that implementation.

manager.log

MDSeries Queue Manager Status					
1	First Queue Item	Workload Balancing Server	3:32 PM - 11/10/99	Bob's Computer	Received
2	Second Queue Item	Workload Balancing Server	3:43 PM - 11/10/99	Greg's Computer	Received
3	Third Queue Item	Workload Balancing Server	3:44 PM - 11/10/99	Bob's Computer	Received
4	Fourth Queue Item	Workload Balancing Server	3:45 PM - 11/10/99	Angela's Machine	Removed
5	Fifth Queue Item	Workload Balancing Server	3:47 PM - 11/10/99	Greg's Laptop	Received
6	Sixth Queue Item	Workload Balancing Server	3:52 PM - 11/10/99	Anthony's Box	Not Received
7	Seventh Queue Item	Workload Balancing Server	4:02 PM - 11/10/99	Greg's Computer	Received
8	Eighth Queue Item	Workload Balancing Server	4:03 PM - 11/10/99	Torrie Server	Queuing

Main idea for disclosure [REDACTED] continued

  
algorithm.jpg



**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**